

Aplicacions en temps real mitjançant l'ús de web sockets en PHP5

Rubén Guix Fernández

Resum– Actualment, l'ús globalitzat de les noves tecnologies ha dut a la societat a una dependència pràcticament exponencial d'aquestes i ha donat pas a l'internet de les coses. Les necessitats i les tendències d'ús mostren com necessàriament l'àmbit del temps real en les aplicacions i la connexió al núvol, és prioritari per a pràcticament qualsevol aplicació d'ús general. En aquest article es detallen les dependències i limitacions a l'hora de solucionar dita problemàtica amb el llenguatge PHP5. Al llarg del projecte, s'ha estudiat la tecnologia per tal de definir i implementar un nucli de temps real en un projecte en producció, solucionant les problemàtiques actuals de dinamisme i intercomunicació d'usuaris. En l'àmbit del projecte, s'ha ampliat la funcionalitat del mateix, mitjançant la implementació d'uns mòduls dissenyats per gestionar en temps real el comerç electrònic de productes online mitjançant l'ús de les subhastes, xats i un observador encarregat de mantenir sempre la pàgina actualitzada a totes les connexions al sistema.

Paraules clau– PHP5, WebSockets, Subhastes, Temps real, pthreads, multi tasca

Abstract– Nowadays, globalized usage of new technologies has led society to an almost exponential dependence on them, giving way to the Internet of Things. Usage trends show how real-time applications and staying connected to the cloud are a priority for virtually any task. This article describes dependencies and limitations when solving this problem using the PHP has been studied throughout the project in order to define and implement a real-time kernel and deploy it into an environment already running in production, solving current problems of dynamism and intercommunication between users. The functionality of the original application has been expanded in the scope of this project, through the implementation of a module designed to manage real-time e-commerce products using auctions, chat rooms and a polling worker keeping all connections up to date at all times.

Keywords– PHP5, WebSockets, Auctions, Real time, pthreads, Multi task

1 INTRODUCCIÓ

LA societat actualment s'està tornant molt dependent de sincronitzar tota la informació, i intenta tenir-hi accés de forma senzilla i ràpida, des de qualsevol punt. Avui en dia es comença a parlar d'un nou concepte, "L'Internet de les coses" (IoT)[1], bàsicament es tracta de connectar-ho tot a internet, per tal de fer-lo accessible en temps real des de pràcticament qualsevol punt del planeta. La tendència actual ens demostra que la sincronització és molt important per a l'usuari i les tecnologies actuals cada cop estan més preparades per dur a terme la solució d'aquests requisits.

En l'actualitat existeixen un gran ventall d'aplicacions on-line que ofereixen el seu servei en temps real. Les tecnologies usades per tal d'implementar el temps real són molt àmplies, depenent directament del grau de maduresa de l'empresa a nivell IT i del seu equip. Aquestes tecnologies poden derivar en costos molt elevats tant a nivell de programari com de maquinari, o bé en el manteniment derivat dels mateixos.

Les empreses petites i mitjanes solen tenir un greu problema en els costos derivats de les llicències utilitzades durant el desenvolupament dels seus productes o serveis tecnològics, tanmateix, els suports físics també veuen incrementat el seu cost i manteniment dependent de la tecnologia de servidor escollida.

El llenguatge de programació PHP5 [9], permet ser executat en la majoria dels servidors i plataformes web actuals sense cap mena de cost. Avui en dia grans llocs

E-mail de contacte: ruben.estudis@gmail.com
Menció realitzada: Enginyeria del Software
Treball tutoritzat per: Gemma Sánchez Albaladejo

d'internet com Facebook han mostrat interès per introduir PHP com a tecnologia de servidor, una bona dada és el fet que es troba instal·lat en més de 20 milions de servidors web. No obstant, no és un llenguatge tan madurat com ho pot ser JAVA i, per tant, molt sovint ens trobem amb limitacions que en ocasions poden portar a la decisió de descartar PHP5 com a tecnologia de servidor.

Amb la finalitat de respondre a un gran volum de qüestions deduïbles de la problemàtica presentada en aquest projecte, es decideix solucionar la necessitat d'una pàgina d'internet sorgida durant el desenvolupament de les pràctiques de l'assignatura: "Tecnologies de Desenvolupament per a Internet i Web" [6].

Actualment aquest espai web, es dedica a proporcionar la plataforma necessària, per a que els particulars o les empreses, puguin realitzar intercanvis comercials online, sense cap intermediari. La problemàtica detectada en aquest espai web es basa en la necessitat de facilitar la comunicació entre els usuaris actius, dinamitzar les accions realitzades, per tal que aquestes es reflecteixin en temps real en les altres connexions actives i així augmentar la proposta de valor actual del lloc web, també s'ha decidit permetre als usuaris realitzar el comerç electrònic mitjançant la modalitat de subhastes. Per tal de respondre als requisits plantejats es realitza el disseny, desenvolupament i integració d'un nucli de software encarregat de gestionar totes les connexions, i capaç de treballar en sincronia amb la plataforma existent.

La figura 1, detalla exactament la ubicació del projecte en el context de l'espai web proposat i els objectius a curt termini del mateix.

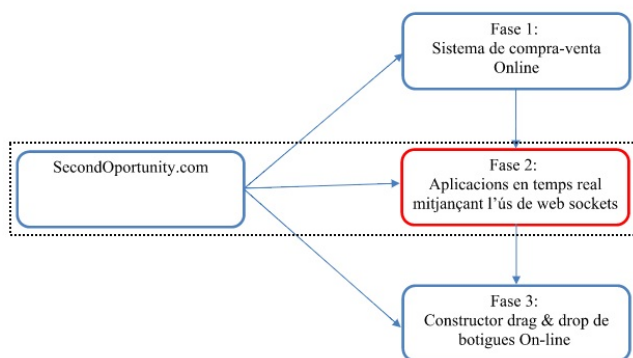


Figura 1: Fases del projecte base

Es pot veure com el projecte es divideix en 3 fases.

- La primera fase, s'orienta en tenir un sistema bàsic, que permeti als usuaris publicar un producte i gestionar el seu compte, a més a més, de la incorporació dels conceptes SEO [7] bàsics. En conclusió és un sistema de producte mínim que permet anar-lo iterant amb la finalitat d'incrementar la seva proposta de valor inicial.
- La segona fase, és la que ubica aquest projecte final de grau, que permetrà una dinamització de la fase 1 i una posterior implementació de la fase 3, essent així un punt d'inflexió en l'aplicació, que marca clarament els objectius de futur de la plataforma.

- La tercera fase permetrà a l'usuari la possibilitat de crear la seva pròpia botiga online mitjançant la tecnologia "Drag & Drop"[8], i mitjançant el nucli desenvolupat en aquest projecte, permetrà la propagació de canvis en temps real, a totes les connexions marcades com actives en el moment en que l'usuari desi els canvis, sense que els altres usuaris necessitin la realització de cap mena d'acció per rebre'ls.

El nucli de temps real, rebrà el protagonisme principal durant tot el desenvolupament posterior, ja que aportarà una integritat al sistema que actualment no té. Per tal d'exemplificar-ho, és necessari pensar, que si un venedor ha realitzat la baixa d'un producte en el moment que un comprador l'està revisant per decidir-se, aquest no se n'adonà que el producte ja no està disponible, fins que realitzi la opció de compra. En canvi, mitjançant el nucli es pot informar en temps real a l'usuari que el producte ja no resta disponible i el perquè. Com a conseqüència es redueix la frustració de l'usuari i s'incrementa la proposta de valor de la plataforma.

Aquest article segueix la següent estructuració. En la Secció 2 es presenten els aspectes del llenguatge PHP5 que s'han considerat més rellevants per dur a terme el projecte. Durant la Secció 3 es detallen els objectius fixats des de l'inici del projecte. La metodologia seguida pel desenvolupament de l'aplicació s'exposa al llarg de la Secció 4. La Secció 5 continuarà tots els detalls relacionats amb els resultats obtinguts en el projecte, en aquest cas, la implementació dels mòduls desenvolupats a partir de l'anàlisi i l'estudi dut a terme de les tecnologies. Finalment la Secció 6 recollirà la conclusió final del projecte, acceptant o no PHP5 com a tecnologia de servidor per dur a terme aplicacions en temps real en un entorn empresarial.

2 ESTAT DE L'ART

En la actualitat la tecnologia de Sockets en PHP, ens permet establir, una comunicació bidireccional i full-dúplex en un sol Socket TCP [25]. La API de WebSockets està essent normalitzada per W3C [4] i el protocol ja va ser normalitzat per la IETF com a RFC 6455 [26]. La necessitat de comunicació constant queda solucionada per part de PHP, no obstant, és necessari que la comunicació sigui òptima. Cal tenir en compte que al tractar-se d'aplicacions client servidor, automàticament s'hereta un temps de latència en el transport d'informació, ja sigui degut a les distàncies físiques entre els punts de connexió o a les limitacions tecnològiques per part dels participants en la connexió (servidor, client o canal de comunicació).

Per tal d'assolir un bon resultat en el concepte d'aplicació en temps real, és necessari tenir en compte, els factors esmentats i realitzar un treball el màxim d'òptim possible. La realització d'aquest objectiu obliga a plantejar com es pot portar a terme la resolució de múltiples peticions de forma concurrent. Aquest últim punt requereix la implementació de fils d'execució que permetin la concurrència entre els processos actius de l'aplicació, que precisament aquesta ha estat la limitació més important trobada en el context d'aquest projecte.

Per tal de realitzar aquesta concurrència en PHP5 s'ha descobert que actualment aquesta funcionalitat es troba en un estat experimental i es fa mitjançant la llibreria Pthreads [10], que intenta simular el comportament dels fils d'execució de JAVA. El problema resideix, en què PHP cada cop que genera un fil d'execució crea un context nou independent, aquesta característica, fa molt complicada la tasca de desenvolupament per permetre la comunicació de dades entre contextos, ja que aquestes han de ser serialitzables.

En el cas de comunicar objectes compartits entre contextos, si aquests no estan en un àmbit local del mateix context, el temps d'execució s'incrementa dràsticament, ja que el que fa és realitzar una còpia de l'objecte i transmetre'l al nou context, sense que aquest pugui ser compartit trivialment amb els demés contextos. A més a més, tenint en compte que l'ús de les matrius actualment és imprescindible en pràcticament qualsevol desenvolupament orientat a objectes, la llibreria presenta la problemàtica, que actualment no permet la serialització d'aquestes estructures de dades.

Tot això incrementa la dificultat a l'hora de decidir-se per PHP com a tecnologia de servidor i enfosca la professionalització del llenguatge en aquest sentit, portant a les empreses a altres alternatives o solucions equivalents en altres llenguatges de programació com per exemple JAVA.

3 OBJECTIUS

En aquest projecte s'han definit uns objectius orientats a la implementació dels coneixements obtinguts en l'estudi de les tecnologies implicades. S'han categoritzat els objectius com a crítics i prioritaris, seguidament es detallen i resumeixen cadascun dels objectius fixats:

- Nucli del sistema mitjançant Sockets i Multitasca:
Aquest objectiu és crític i es basa en el disseny i implementació d'un nucli de sistema capaç de treballar en sincronia amb la plataforma actual que hi ha en producció.
- Entorns:
Objectiu prioritari per tal de poder portar el desenvolupament d'una forma segura i seguint unes pautes de treball el màxim de professionals possibles, aquests entorns són: l'entorn de desenvolupament, pre-producció i producció.
- Proves:
Objectiu prioritari que integra el concepte de proves automàtiques al projecte fent servir algun sistema o llibreria que permeti l'automatització.
- Capa d'abstracció a base de dades:
Objectiu prioritari degut a la necessitat de fer l'accés a la base de dades d'una forma estandaritzada mitjançant la selecció i implementació d'una capa d'abstracció senzilla i segura.

- Desenvolupament dels mòduls de resultats:

Per tal de mostrar els resultats obtinguts en l'estudi de les tecnologies implicades s'ha decidit dissenyar i implementar els següents tres mòduls.

- Mòdul de subhasta: Aquest serà el mòdul encarregat de proporcionar una nova modalitat de comerç electrònic en la plataforma mitjançant la comunicació en temps real.
- Mòdul observador:
El bloc observador serà el mòdul encarregat de mantenir la plataforma actualitzada per a totes les connexions actives, per tant tots els canvis realitzats en els articles que formin part de la modalitat de subhastes seran actualitzats a la pàgina principal, informant als visitants de quin és el preu i el guanyador actual sense que sigui precisa cap acció per part de l'usuari.
- Mòdul d'intercomunicació:
Aquest bloc serà capaç de facilitar la comunicació entre usuaris mitjançant un xat en la plataforma, també serà l'eina clau per posar en contacte compradors i venedors en la modalitat de subhasta, ja que aquestes seran anònimes i, per tant l'usuari comprador no sabrà qui és l'usuari venedor. Només l'usuari guanyador tindrà el contacte de l'usuari propietari de la subhasta un cop aquesta estigui en un estat de finalitzada.

- Integració en la plataforma:

Aquest és un objectiu prioritari en el que s'haurà d'analitzar l'estat actual de la plataforma per posteriorment dissenyar i implementar els canvis per tal de dur a terme la integració dels mòduls desenvolupats.

4 METODOLOGIA

En aquest apartat es posa de manifest la metodologia seguida durant el transcurs del projecte, s'estructura amb els punts més remarcables com són: la inicialització del projecte, el disseny i la implementació dels resultats obtinguts.

4.1 Inicialització del projecte

En aquesta secció s'introduiran i seran explicades les tecnologies utilitzades durant el projecte tant en l'àmbit de gestió com de desenvolupament. També es fa incís en l'estructura del projecte i els entorns de desenvolupament.

4.2 Selecció de tecnologies

Per a la realització del projecte, s'ha realitzat la tasca d'anàlisi de les tecnologies necessàries per tal d'acomplir els objectius fixats en el projecte. L'Objectiu principal a solucionar es la necessitat de que el sistema tingui un caràcter el més proper possible al temps real. Degut que el projecte s'enmarca en el camp de l'anàlisi del llenguatge PHP5, la documentació consultada ha estat la oficial d'aquest llenguatge. Cal dir que PHP disposa d'una documentació i una comunitat molt extensa que permet la cerca d'informació provinient de diferents fonts i així poder anar consultant i contrastant sense cap complicació.

4.2.1 Tecnologies de gestió

S'ha realitzat una tasca de selecció d'eines necessàries per a la realització del projecte a nivell de gestió, a continuació es detallen les solucions que s'han cregut més òptimes.

- Gantter [27]: L'aplicació de Google Drive "Gantter" és l'eina que ens permetrà dur a terme la planificació mitjançant diagrama de Gantt de forma dinàmica, és a dir, sense dependre de software privatiu ja que només precisa d'una connexió a internet i un navegador.
- Bitbucket [20]: és un repositori basat en git que ens permetrà mantenir el desenvolupament sincronitzat en tots els equips i a més a més ens permetrà realitzar tasques de manteniment i integritat de còdi.
- Bitbucket Cards [21] permetrà realitzar una integració de la gestió en el desenvolupament, dita integració permetrà treballar de forma sincronitzada a nivell de tasques i poder portar un log de les tasques realitzades i les observacions que vagin sorgint durant el procés, també és òptim a l'hora de pensar en les dependències, que a l'hora de planificar el projecte no hagin estat contemplades.

4.2.2 Tecnologies de desenvolupament

La primera tecnologia a investigar era com PHP soluciona l'aspecte de la comunicació constant entre el client i el servidor. Per tal de solucionar-ho les alternatives investigades han estat orientades als web sockets per tal d'obtenir una connexió full-duplex [28], persistent i bidireccional. La filosofia passa per mantenir una connexió directa oberta fins que el client o el servidor decideixen finalitzar-la. Durant el temps en que la connexió està activa, el client i el servidor es poden enviar missatges directament sense necessitat de tornar a establir les connexions, i el servidor en tot moment és conscient de l'estat de totes les connexions que té establertes, de tal manera que les pot gestionar utilitzant d'una forma òptima els recursos dels que disposa, en la figura 2 es pot observar el procés descrit.

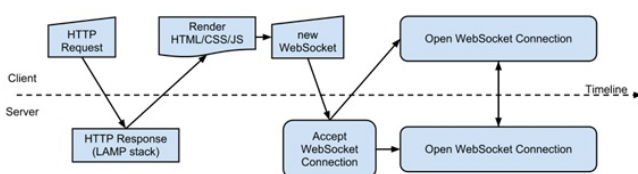


Figura 2: Procés de comunicació WebSocket

En l'extrem del client els web sockets estan integrats de forma nativa als navegadors Firefox, Opera i Safari (incloent Safari mòbil), com a dada considerable podem dir que els navegadors que no acceptin aquestes tecnologies poden fer ús del plugin Flash Polyfill [11]. Entre d'altres la llibreria Ratchet [12] semblava la més interessant ja que tenia molt madurat aquest tema, però degut a que la motivació de realitzar aquest treball passa per aprofundir més en la tecnologia de web sockets, aquesta ha estat descartada per tal de poder realitzar una implementació pròpia de webs sockets i poder introduir el concepte de

multi tasca que s'explicarà seguidament.

Un altre aspecte detectat, és la concurrència, les aplicacions en temps real no només han de tenir un connexió constant sinó que també han de poder realitzar tasques concurrentment entre fluxos de treball no dependents. A l'hora de realitzar la selecció de tecnologies es descobreix que PHP, aquest aspecte no el té nativament implementat. Com a solució existeix una llibreria de PHP en una fase bastant avançada d'experimentació anomenada pthreads [10] basada en POSIX [29]. La llibreria pthreads incorpora a PHP, la possibilitat de realitzar fils d'execució donant solució per tant, a la necessitat de concurrència del projecte. L'habilitació dels entorns de desenvolupament ha estat la part responsable de gran part de les variacions en la planificació i execució del projecte, degut a la necessitat d'investigar i assolir els coneixements necessaris per tal d'integrar, les dependències necessàries en els entorns corresponents.

Amb la finalitat de no dedicar recursos a aspectes que no entren en l'àmbit del projecte, però sí són necessaris per tal d'acomplir els requisits de les implementacions de resultats, s'ha decidit incorporar les següents llibreries:

- JQuery [13]: És una llibreria que facilita la tasca d'implementar Javascript en el costat del client i permetrà accés als membres del DOM [19] sense gaire complicació, ja que l'aprenentatge de la llibreria és ràpida i permet la implementació de gran varietat de solucions. Actualment hi ha solucions desenvolupades per pràcticament totes les necessitats que poden sorgir pel que fa a les funcionalitats implementades en el client.
- Powerange [14]: Un dels requisits durant la implementació del mòdul en temps real de les subhastes és que el client pugui decidir quan vol incrementar la subhasta dins d'uns límits establerts per diversos factors que defineix l'estat de la subhasta. Aquesta llibreria es basa en JQuery, i permet que mitjançant una barra desplaçable per l'usuari, aquesta envii el valor escollit per l'usuari a la funció definida en el servidor client i d'aquesta manera enviar-lo al servidor PHP per tal de ser processat conjuntament amb els altres valors de la subhasta. En la figura 3 es pot veure el resultat del complement implementat en el projecte.
- JQuery TimeCircles [15]: És un plugin de JQuery que resoldrà la necessitat que l'usuari de les subhastes tingui d'una forma atractiva i visual la informació del temps restant de la subhasta.
- PHP - PDO [22]: L'objecte natiu PDO de PHP, ens permetrà complir amb la tasca d'implementar una capa d'abstracció a base de dades, de forma senzilla i amigable per al futur manteniment.

Tota aquesta sèrie de recursos ens permeten enfocar-nos en el context real del projecte, sense temor de que la implementació dels resultats sigui pobre en disseny o pugui afectar a la integritat del fluxe de treball, ja que són solucions molt testades prèviament per la comunitat. En les figures 3 i 4 es pot veure els resultats dels complements implementats en el projecte.



Figura 3: Powerange JQuery en el mòdul de subhastes



Figura 4: TimeCircles JQuery

- Selenium IDE [18]: És un entorn de desenvolupament integrat complet (IDE) per a les proves de selenium i s'implementa com una extensió de Firefox. Permet l'enregistrament, edició i depuració de proves. Selenium IDE inclou tot el nucli de Selenium, permetent als desenvolupadors gravar i reproduir les proves de forma fàcil i ràpida en un entorn realista en el que s'executarà. No només és una eina d'enregistrament, sinó un IDE complet. Les característiques de Selenium IDE inclouen:

- Enregistrament i reproducció senzilla,
- Selecció de camps intel·ligent fent servir IDs, noms o XPath [30]
- Autocompletat per a totes les comandes de Selenium,
- Debug i proves amb breakpoints
- Guardar tests com HTML
- scripts de Ruby o qualsevol altre format
- Fàcil ús d'extensions
- Captura de pantalla automàtica en cas d'error

4.3 Estudi de les tecnologies escollides

Al ser tecnologies poc conegudes per l'estudiant s'ha decidit dur a terme l'anàlisi per separat de cadascuna d'elles i la implementació en format de petits projectes per tal de conèixer i garantir la coherència entre totes elles.

De totes les tecnologies escollides només una ha donat problemes fins al punt de contemplar la possibilitat de descartar la implementació en temps real mitjançant PHP5 ja que un objectiu bàsic en aquest projecte és poder executar en paral·lel un conjunt d'instruccions d'un programa i així aproximar-nos al màxim al concepte de temps real. La implementació de la llibreria pthreads [10] ha estat una tasca complexa degut a que té limitacions a nivell de sincronitzar algunes estructures de dades, com els vectors entre fils d'execució concurrents. Seguidament s'explica el concepte i la llibreria.

El servidor utilitzat en la implementació del projecte disposa de diversos nuclis, per tant permet executar diversos fils d'execució del mateix procés de forma paral·lela, resultant en una millora potencial en el rendiment. En concret passem de l'execució seqüencial del programa a l'execució de blocs en paral·lel sincronitzant en certes ubicacions del programa els fils d'execució per tal que els resultats de l'execució siguin els correctes.

Un aspecte interessant és que els fils d'execució comparteixen la mateixa regió de memòria que el procés que els va iniciar, per tan diversos fils poden treballar en la mateixa memòria de dades, no obstant això, és important controlar l'accés a les dades quan aquestes poden ser variades, ja que aquest accés pot ser concurrent.

La llibreria Pthreads disposa de la classe Thread, que és l'encarregada de crear el fil d'execució, en ser una classe abstracta ha de ser implementada per una classe filla, en el nostre cas aquesta classe és el mòdul encarregat d'escollar les connexions establertes (ClientListener.php). La classe Thread ofereix múltiples mètodes que no poden ser substituïts en les classes filles. Un mètode abstracte anomenat run que ha de ser implementat obligatòriament, serà el nucli del nostre mòdul encarregat d'escollar les connexions ja establertes, i reaccionar en conseqüència aplicant la lògica de programa corresponent.

```
C:\xampp\htdocs\testingThread>php TestingThread.php
Thread: 6 Temps: 1 segons
Thread: 5 Temps: 1 segons
Thread: 8 Temps: 1 segons
Thread: 3 Temps: 2 segons
Thread: 2 Temps: 2 segons
Thread: 7 Temps: 3 segons
Thread: 4 Temps: 3 segons
Thread: 1 Temps: 4 segons
Thread: 0 Temps: 4 segons
Thread: 9 Temps: 4 segons
```

Figura 5: Test d'execució pthreads

En la figura 5 es pot observar el test realitzat per tal de demostrar que la llibreria fa el que la teoria de fils d'execució marca. La prova executada es tracta que cada fil d'execució selecciona un temps a l'atzar i després d'esperar el temps assignat, imprimeix el missatge de sortida. Com ha resultat es veu que tot i ser adjudicat aleatòriament el temps a cada fil d'execució, finalitzen primer els que tenen el període de temps més curt.

Com es pot observar les execucions pràcticament s'executen al mateix temps. Com que es disposa de 8 processadors en la màquina on s'ha executat la prova, es pot apreciar com els fils d'execució s'han anat executant depenent de la disponibilitat d'ús gestionada pel sistema operatiu.

4.4 Estructura del projecte

El projecte es divideix seguint el patró de disseny model·vista-controlador (MVC) [23], com podem veure a la figura 6 tenim la carpeta controller on s'observa la classe AuctionServer, dita classe és l'encarregada de contenir les funcionalitats comuns tant per les subhastes com pel sistema d'intercomunicació d'usuaris. D'aquesta forma

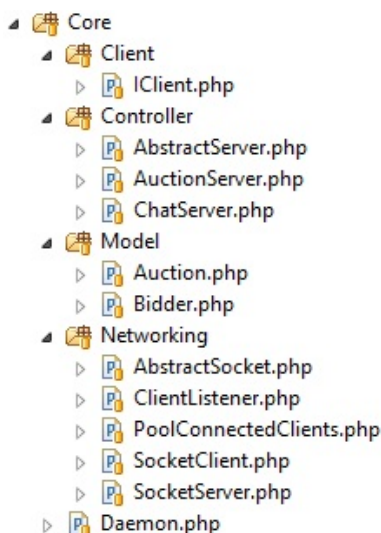


Figura 6: Estructura del projecte

obtenim una coherència d'orientació a objectes mitjançant l'ús del polimorfisme. Els servidors per a les subhastes i el sistema d'intercomunicació són les classes AuctionServer i ChatServer. Aquestes classes gestionaran totes les peticions corresponents a cada tipologia de petició. Com a detall en la figura 6 només es contempla la part del nucli, per tant la vista no apareix perquè la ubicació de la mateixa resideix en l'estructura de directoris pròpia de la plataforma.

A nivell de model es poden observar dos classes, Auction i Bidder. Aquestes classes són bàsicament classes Object Relational Mapping (ORM) [24] que fan la funció de representar en format d'objecte el contingut de la base de dades.

Per altra banda hi ha un mòdul de Networking que conté tot el relacionat amb la comunicació client-servidor mitjançant web sockets. Seguint la mateixa filosofia la classe AbstractServer conté les operacions genèriques, i les classes SocketServer i SocketClient les específiques. El disseny s'ha realitzat pensant en la tercera fase del projecte base que es pot veure en la figura 1, per tal que en un futur es pugui realitzar una API que permeti a la tercera fase fer ús de tot el potencial del nucli.

Cal destacar que la classe més important d'aquest mòdul és PoolConnectedClients, ja que és la classe encarregada de mantenir la llista de tots els clients connectats a la plataforma i així permetre, que els diferents fils d'execució del sistema tinguin accés a la llista mitjançant la crida de la propietat de la classe (àlies de l'usuari) per recuperar el valor (recurs Socket del sistema). El problema més important que s'ha apreciat en aquest projecte ha estat la complexa tasca duta a terme per tal d'intercomunicar la informació precisada pels contextos.

La llibreria pthreads de PHP comparteix la informació a través de serialització de les dades dels objectes, en el cas de les cadenes o objectes de tipus natiu els serialitza sense cap problema, els arrays nadius en php no permeten la serialització.

Per tal de solucionar aquest problema es contemplen varies solucions, entre elles, el fer ús de base de dades per tal d'emmagatzemar la informació necessària o l'ús de la memòria cau simulant la solució anterior.

Les solucions esmentades no eren òptimes degut a la necessitat d'anar generant les consultes a base de dades o a memòria suposava una latència no assumible en la segona fase del projecte "Subastas", en la primera fase "Chat" al tractar-se d'una latència mínima no hagués suposat cap mena de problema, més enllà de l'ús desproporcionat de recursos (quotes de disc, accessos a la base de dades, quotes de transmissió de dades etc...). Finalment l'opció adoptada és la de crear un objecte de tipus Threaded, el qual es una extensió de Thread que forma part de la llibreria pthreads, de manera que aprofitant l'orientació a objectes de PHP 5, que permet generar propietats dinàmicament dintre d'un objecte afegint entrades en la taula de propietats interna, permetent que aquest objecte es pugui serialitzar, ja que les entrades de la taula de propietats, són de tipus cadena o primitives, el qual ens permet intercanviar informació entre els diferents Threads que cal recordar que són diferents contextos d'aplicació.

Cal remarcar que cada vegada que es crea un objecte de tipus Threaded també es crea el seu propi context, per tant cada vegada que s'accedeix a les propietats d'un objecte es genera una connexió entre contextos per tal d'intercanviar informació. En la figura 7 es pot veure l'exemplificació de lo esmentat.

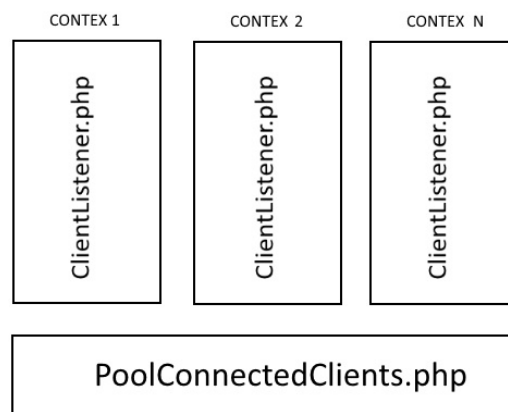


Figura 7: Sistema de comunicació entre contextos

Mitjançant la classe PoolConnected Clients, el fluxe de treball mostrat en la figura 8 per tal d'afegir, eliminar o consultar usuaris s'ha simplificat molt. La problemàtica resolta podria ser que el context 1 hagi ofert un canvi, per exemple en una propietat interna d'un client. Com aquest canvi ha estat realitzat en el context 1 el canvi no ha estat reflectit en el context 2, per tal d'enviar el canvi que s'ha produït en el context 1 aquest fa ús de la PoolConnectedClients i així serialitzar el canvi i que pugui estar disponible per quan el context 2 necessiti verificar la propietat d'aquest client.

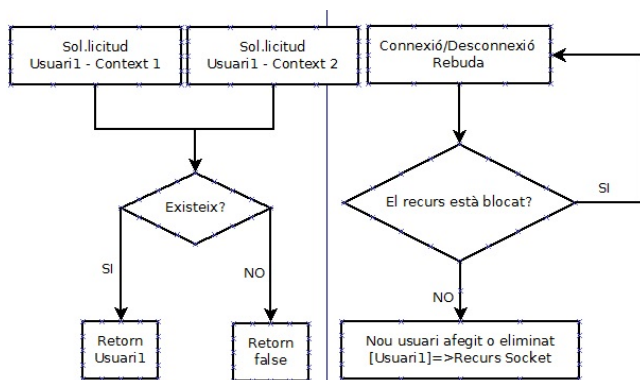


Figura 8: Procés de connexió/desconnexió

4.5 Habilitació dels entorns

L'ús de les tecnologies mencionades van lligades a uns requisits a nivell de servidor que han estat tinguts en compte a l'hora d'implementar els entorns de desenvolupament. Per tal d'adaptar la metodologia a una forma de treballar professional s'ha optat per tenir tres entorns, un de desenvolupament amb Windows ubicat a la màquina personal i uns altres dos idèntics amb Linux que són els de preproducció i producció. Aquests dos últims entorns han estat clau a l'hora de garantir la migració del codi generat, a l'entorn de producció. Primerament un cop desplegada l'aplicació en l'entorn de desenvolupament aquesta es passa a l'entorn de preproducció que té les mateixes característiques que el de producció i finalment es desplega conjuntament amb la plataforma en el entorn de producció per tal d'evitar caigudes innecessàries o pèrdua de dades.

4.5.1 Entorn de desenvolupament

Aquest entorn ubicarà totes les proves de les tecnologies escollides i el desplegament instantani del codi generat per tal d'agilitzar les proves i les comprovacions sense que aquestes requereixin gaire temps per a ser executades i així afavorir el temps productiu durant el desenvolupament. Aquest entorn consta d'un sistema operatiu Windows 8.1 Pro i una màquina AMD FX(tm)-8350 Eight-Core Processor 4.00 Ghz. Amb 8 GB de RAM, MySql 5.6 (motor de base de dades INNODB) i PHP 5.6.

Per tal de poder executar la llibreria multitasca, ha estat precis la incorporació de dues llibreries: `php.pthreads.dll` al directori `ext` de `php` i `pthreadVC2.dll` al directori de `php` per tal de poder activar la funcionalitat indicant l'extensió `php.pthreads` al fitxer de configuració de PHP `php.ini` ja que Windows no incorpora l'especificació POSIX [29]

4.5.2 Entorn de pre-producció i producció

S'ha cregut necessari realitzar aquest apartat amb els dos entorns junts, ja que són idèntics, bàsicament necessaris degut a que actualment hi ha una aplicació en producció, i per tal d'evitar errades que puguin afectar a la producció normal del lloc web, primer, s'han testejat totes les aplicacions en un entorn previ de pre-producció.

Aquests entorns estan ubicats en un sistema cloud de servidors propietat de l'empresa Amazon mitjançant el seu

servei anomenat Amazon EC2 [16]. Per tal de mantenir els costos del projecte sota mínims s'ha contractat el servei de micro instància, ja que proporciona durant un any el servidor dedicat sense cap cost. Aquest servidor consta de les següents característiques: Processadors Intel Xeon d'alta freqüència que funcionen a 2,5 GHz amb Turbo fins a 3,3 GHz, 1GB de RAM amb un sistema operatiu Ubuntu 14.04 Server.

Els entorns han estat accessibles des de l'exterior durant el desenvolupament del projecte mitjançant les següents adreces:

- Entorn de pre-producció: test.secondoportunity.com
La instància d'aquest entorn al ser una segona instància micro de Amazon ha estat eliminada un cop testat el sistema degut a la no gratuïtat de la mateixa
- Entorn de producció www.secondoportunity.com
Aquest entorn és accessible, ja que està ubicat en la instància actual de la plataforma.

Per tal de poder fer ús de les característiques de multi tasca que proporciona la llibreria `pthread`, la preparació dels entorns en Linux no ha estat tant trivial com a Windows. Ha estat necessària la versió PHP-Zts, ja que és un requeriment a l'hora d'executar `pthread` [10]. Per poder realitzar la compilació de PHP ha de tenir activats els flags de la llibreria, a més a més d'identificar tots els mòduls necessaris per a mantenir la funcionalitat actual del sistema en producció. Per tal d'evitar problemes de dependències durant la implementació del projecte

5 RESULTATS

5.1 Modul Intercomunicació

El flux de treball realitzat per dur a terme el mòdul d'intercomunicació passa per tenir a l'usuari prèviament autenticat en la plataforma, si aquest usuari no està autenticat no té sentit que pugui agregar contactes a la seva llista. Per tant l'usuari una vegada autenticat, realitza una cerca de l'usuari que vol afegir a la llista, a mesura que va escrivint l'hi apareixen els resultats en temps real dels àlies d'usuari que contenen la cadena que l'usuari està escrivint per teclat. Un cop la llista mostra l'usuari que s'està cercant, només cal seleccionar-lo i acceptar el missatge de confirmació que apareix per pantalla. Un cop l'usuari té el seu contacte a la llista, tal com es pot observar en la figura 10, només ha de realitzar un clic sobre l'usuari de la llista i automàticament se l'hi obre una finestra de xat amb la que pot començar a parlar. Cal destacar que a l'usuari final no se l'hi obre la finestra fins que l'usuari interessat en parlar no l'hi envia el missatge per tant s'evita que a l'obrir una finestra nova per error l'altre usuari sàpiga que l'usuari inicial ha intentat inicialitzar una conversa amb ell. Per tal de portar a terme la integració del mòdul d'intercomunicació s'ha realitzat un anàlisi previ per localitzar una ubicació òptima dins de la plataforma actual. S'ha decidit situar el mòdul just a sota del menú, inspirat directament en el disseny de Google com es pot observar en la figura 10. En la figura 9 podem observar el previ disseny de la finestra de xat i la implementació final en la plataforma.



Figura 9: Disseny i implementació xat

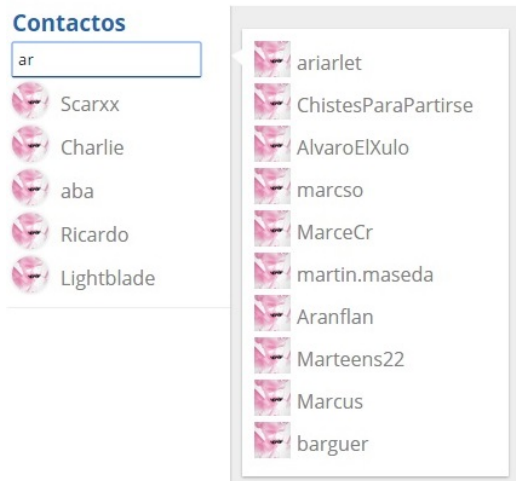


Figura 10: Contactes de l'usuari

5.2 Mòdul de subhastes

Per tal de realitzar el mòdul de subhastes s'ha tingut en compte tot el flux de treball implementat actualment en la plataforma, adaptant una nova modalitat per tal que les subhastes es poguessin contemplar com un tipus de producte diferent als existents. Per tant la primera tasca a realitzar ha estat la detecció dels punts involucrats, tant a nivell d'usabilitat com d'estructura existent en el procés de donar d'alta un producte. Un cop assolida amb èxit aquesta tasca s'ha procedit ha generar un disseny previ de les subhastes, a partir de la plantilla de producte existent en la plataforma, s'ha realitzat una modificació mitjançant un pre-disseny i intentant aprofitar al màxim els espais disponibles, també s'ha tingut molta cura amb la informació que realment es te que mostrar en les subhastes. El disseny previ es pot veure en la figura 11 i el resultat de la implementació en la plataforma es visualitza en la figura 12.

Com es pot veure en la figura 12, en el comptador enrere de la subhasta es pot veure la implementació de la llibreria de JQuery TimeCircles [15] que mitjançant les dades ubicades en la base de dades calcula el temps restant de la subhasta i el mostra d'una forma atractiva i clara per l'usuari, sense una tasca prèvia de disseny, ja que aquesta està ben estipulada per la mateixa llibreria. Cal destacar que la llibreria prèviament ha estat molt testada per part de la comunitat i per tant dona unes garanties d'estabilitat i integritat.

Si ens fixem en la barra desplaçable ubicada en la part inferior del botó en la figura 12, veiem la implementació de la llibreria de JQuery Powerage [14], aquesta igual que en l'anterior, mostra els seus valors de límit mínim i màxim

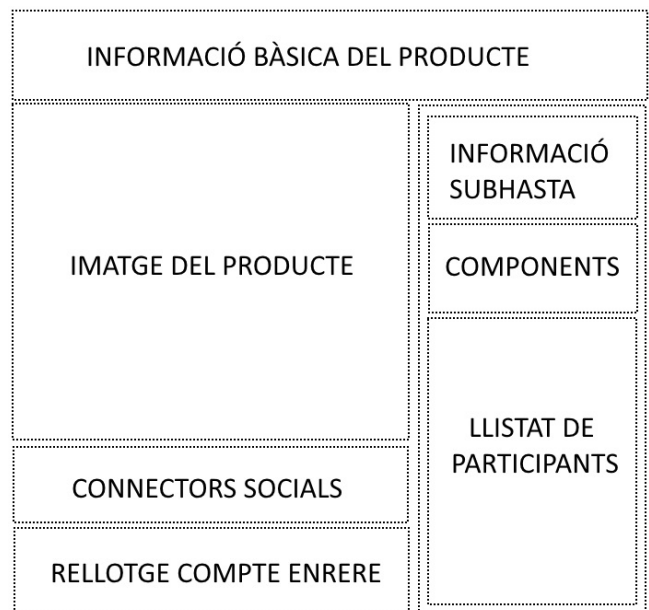


Figura 11: Disseny de la Subhasta

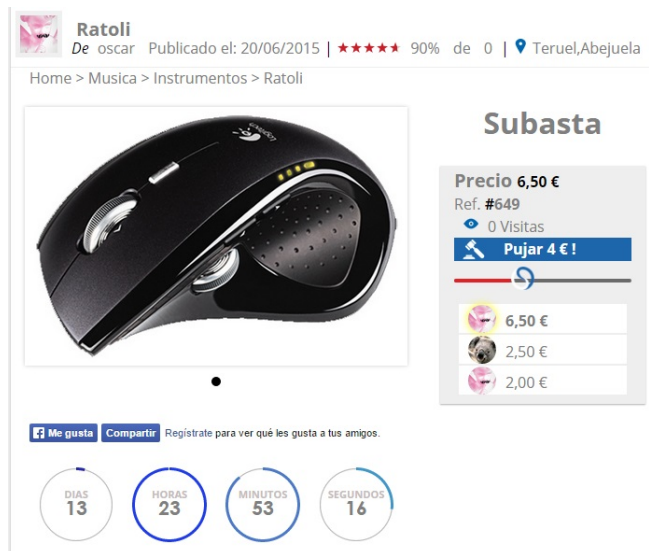


Figura 12: Implementació de la Subhasta

mitjançant la informació actual desada en base de dades, si el preu és inferior a 10 euros mostrarà dos decimals i si és superior el mostrarà només en enters, mitjançant l'opinió dels companys d'universitat s'ha cregut que aquesta seria l'opció més òptima. Dita barra també ha estat modificada per tal de mostrar el logotip de la plataforma i d'aquesta forma realitzar una integració tenint en compte la imatge corporativa de la plataforma. En la part inferior de la barra podem veure el llistat dels usuaris actius en la subhasta ordenats de més preu a menys donant la sensació de ranking, per tal de donar una mica més de protagonisme al guanyador actual de la subhasta la seva imatge es mostra amb una ombra daurada i el preu en negreta.

El flux de treball per a realitzar una subhasta passa pel control de si l'usuari està autenticat en aquell moment o no, és a dir si l'usuari acciona el botó de realitzar acció de subhasta i aquest no està autenticat s'ha decidit redirigir a l'usuari directament a la finestra emergent de login. En

el cas d'estar autenticat en la plataforma l'únic que ha de fer és seleccionar un increment de preu i pressionar el botó d'acció de la subhasta d'aquesta forma l'usuari veurà com la seva imatge, passa a ser la primera de la classificació indicant-li que actualment és el guanyador de la subhasta. Si ens fixem en l'altre extrem que seria un altre usuari que està observant en el mateix instant de temps la subhasta, sense que hagi de realitzar cap mena d'acció, veurà també actualitzat el rànquing. Com a detall de flux de treball cal destacar que des de la classe ClientListener s'emmagatzemen els visualitzadors de les subhastes per tant el servidor en tot moment sap qui està en una sala de subhasta i qui no, per tant no realitza un broadcast a tots els usuaris, l'únic que fa és enviar l'actualització de subhasta als usuaris que han accedit a la sala de la subhasta en qüestió. D'aquesta forma s'aconsegueix una reducció considerable en l'ús de recursos del servidor.

5.3 Mòdul Observador

El mòdul observador és l'encarregat d'aportar dinamisme a la plataforma, per tant aquest mòdul tindrà un gran protagonisme a partir d'ara en tots els desenvolupaments futurs, ja que podrà ser implementat en qualsevol punt de la plataforma. Per tal de mostrar el resultat s'ha decidit realitzar un canvi en els blocs de productes relatius a les subhastes de la pàgina principal de la plataforma, d'aquesta forma els usuaris que tinguin la pàgina principal oberta veuran els preus actualitzats de les subhastes en temps real i la icona dels usuaris que les estan guanyant en tot moment. En la figura 13 es mostra el predisseny d'un bloc individual amb el contingut que s'ha cregut rellevant per l'usuari i posteriorment en la figura 14 es veu el resultat de la implementació.

Com es pot observar en la figura 14 tenim quatre productes actius amb les seves corresponents imatges i preus, els preus s'actualitzen automàticament quan un usuari l'incrementa des d'una sala de subhasta. Si ens fixem en les icones de la part superior esquerra, podem veure com hi ha dos tipus d'icones, un que és un martell, que el que pretén és fer que l'usuari dedueixi que és una subhasta i que encara ningú ha incrementat el preu. Quan un usuari realitza un increment de preu l'observador s'encarrega d'actualitzar aquesta icona i assignar-li la imatge de perfil de l'usuari que en aquest moment està guanyant la subhasta. Per tant a simple vista veiem com en els quatre productes que es mostren en la figura 14 el primer i l'últim encara no tenen cap increment del preu inicial i el segon i tercer són subhastes que ja han sofert un increment, i per tant veiem icones corresponents a les imatges de perfil dels usuaris que estan guanyant en aquest instant de temps la subhasta.

6 CONCLUSIONS

S'han assolit els objectius fixats en la planificació del projecte i es pot afirmar que la posterior implementació dels resultats en la plataforma en producció ha aportat un valor afegit als usuaris. Mitjançant els resultats de la intercomunicació s'ha socialitzat l'espai, amb la nova modalitat de subhasta tot i no ser un projecte empresarial s'ha obert una nova línia de negoci i amb el mòdul observador

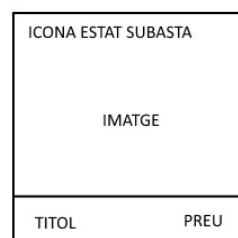


Figura 13: Diseny de la presentació del producte



Figura 14: Presentació de 4 anuncis en la pàgina principal

s'ha incrementat considerablement el dinamisme de l'espai web.

L'estat d'experimentació de la llibreria pthreads no proporciona la confiança necessària per ser utilitzada en un entorn de producció empresarial. Cal destacar que cada cop més s'aproxima al que realment es necessita en entorns multi tasca en l'àmbit empresarial.

AGRAÏMENTS

En primer lloc volgués agrair a Gemma Sánchez l'oportunitat que m'ha brindat per guiar-me en la realització d'aquest projecte, i al Departament d'Enginyeria del Software per permetre'm l'opció, de realitzar el meu treball final de grau en el context d'un projecte personal.

Als meus pares, perquè han patit tant o més que jo durant aquests 4 magnífics anys, ells m'han guiat, donat suport i de ben segur sense ells no hauria finalitzat amb èxit aquesta etapa. I per descomptat a la resta de la meua família, la meua àvia, els meus oncles, cosins,... per la confiança.

A tots els meus professors, des del col·legi fins a la universitat, per tot el que he après gràcies a ells.

A tots els meus companys i amics de la universitat, sobretot a Aureli Bisbe, Esther Piqueras, Francisco Cobo i

Jerónimo Fernandez, Tomàs Artés, Laura Martinez, perquè sense tots vosaltres, els vostres resums, els vostres consells, els “TOP DOWNS” i la vostra ajuda, segur que no estava escrivint ara mateix aquestes línies.

Un cop vaig llegir que els amics són, els que quan plou comparteixen amb tu el paraigües, però si no tenen paraigües comparteixen amb tu la pluja. Per tant agraeixo a Óscar Romero l'ajuda constant i el patiment compartit durant el transcurs de les diferents etapes, projectes junts i sobretot l'hi agraeix-ho el estar al meu costat durant el transcurs d'aquest treball que ens ha portat de cap.

Per últim vull agrair la paciència i el suport rebut per part de la meua parella Lidia Domingo, ja que sense ella de ben segur el camí hagués estat infinitament més difícil.

Gràcies.

REFERÈNCIES

- [1] Internet of Things. https://en.wikipedia.org/wiki/Internet_of_Things (Consultat Juny 2015)
- [2] Comptador de dispositius connectats a internet actualment i previsió fins a 2020. <http://newsroom.cisco.com/ioe> (Consultat Juny 2015)
- [3] Network socket. https://en.wikipedia.org/wiki/Network_socket (Consultat Abril 2015)
- [4] World Wide Web Consortium (W3C). <http://www.w3c.es/> (Consultat Abril 2015)
- [5] The WebSocket Protocol. <http://datatracker.ietf.org/doc/rfc6455/> (Consultat Abril 2015)
- [6] Guia Docent Tecnologies de desenvolupament per a internet i web. <http://www.uab.cat/guiesdocents/2014-15/g102742t2502441a2014-15iCAT.pdf> (Consultat Juny 2015)
- [7] Conceptos de SEO. [https://msdn.microsoft.com/es-es/library/ff723998\(v=expression.40\).aspx](https://msdn.microsoft.com/es-es/library/ff723998(v=expression.40).aspx) (Consultat Maig 2015)
- [8] drag and drop https://es.wikipedia.org/wiki/Arrastrar_y_soltar (Consultat Maig 2015)
- [9] PHP. <https://es.wikipedia.org/wiki/PHP> (Consultat Maig 2015)
- [10] Pthreads PHP. <http://pthreads.org/> (Consultat Maig 2015)
- [11] Polyfill. <https://en.wikipedia.org/wiki/Polyfill> (Consultat Maig 2015)
- [12] Ratchet WebSockets for PHP. <http://socketo.me/> (Consultat Maig 2015)
- [13] JQuery. <https://jquery.com/> (Consultat Maig 2015)
- [14] Powerange. <https://github.com/abpetkov/powerange> (Consultat Maig 2015)
- [15] JQuery TimeCircles. <https://plugins.jquery.com/timecircles/> (Consultat Juny 2015)
- [16] Amazon EC2. <http://aws.amazon.com/> (Consultat Juny 2015)
- [17] Requeriments Pthreads POSIX. <http://php.net/manual/en/pthreads.requirements.php> (Consultat Maig 2015)
- [18] SELENIUM IDE <http://www.seleniumhq.org/projects/ide/> (Consultat Juny 2015)
- [19] Document Object Model. https://es.wikipedia.org/wiki/Document_Object_Model (Consultat Juny 2015)
- [20] Bitbucket. <https://bitbucket.org/> (Consultat Abril 2015)
- [21] Bitbucket Cards. <http://www.bitbucketcards.com/> (Consultat Maig 2015)
- [22] PHP Data Objects. https://es.wikipedia.org/wiki/PHP_Data_Objects (Consultat Maig 2015)
- [23] Modelo-vista-controlador <https://es.wikipedia.org/wiki/Modelo-vista-controlador> (Consultat Juny 2015)
- [24] Mapeo objeto-relacional. https://es.wikipedia.org/wiki/Mapeo_objeto-relacional (consultat Maig 2015)
- [25] Transmission Control Protocol. https://es.wikipedia.org/wiki/Transmission_Control_Protocol (consultat Juny 2015)
- [26] RFC 6455. <https://datatracker.ietf.org/doc/rfc6455/> (consultat Maig 2015)
- [27] Gantter Google Drive Integration. <http://www.gantter.com/google-drive/> (consultat Maig 2015)
- [28] Dúplex (dúplex completo o full duplex). [https://es.wikipedia.org/wiki/Dplex_\(telecomunicaciones\)](https://es.wikipedia.org/wiki/Dplex_(telecomunicaciones)) (consultat Maig 2015)
- [29] Portable Operating System Interface <https://es.wikipedia.org/wiki/POSIX> (consultat Maig 2015)
- [30] XML Path Language <https://es.wikipedia.org/wiki/XPath> (consultat Maig 2015)

A.2.1 Networking

El diagrama mostrat en la figura 16 fa referència a la part del nucli responsable de la comunicació.

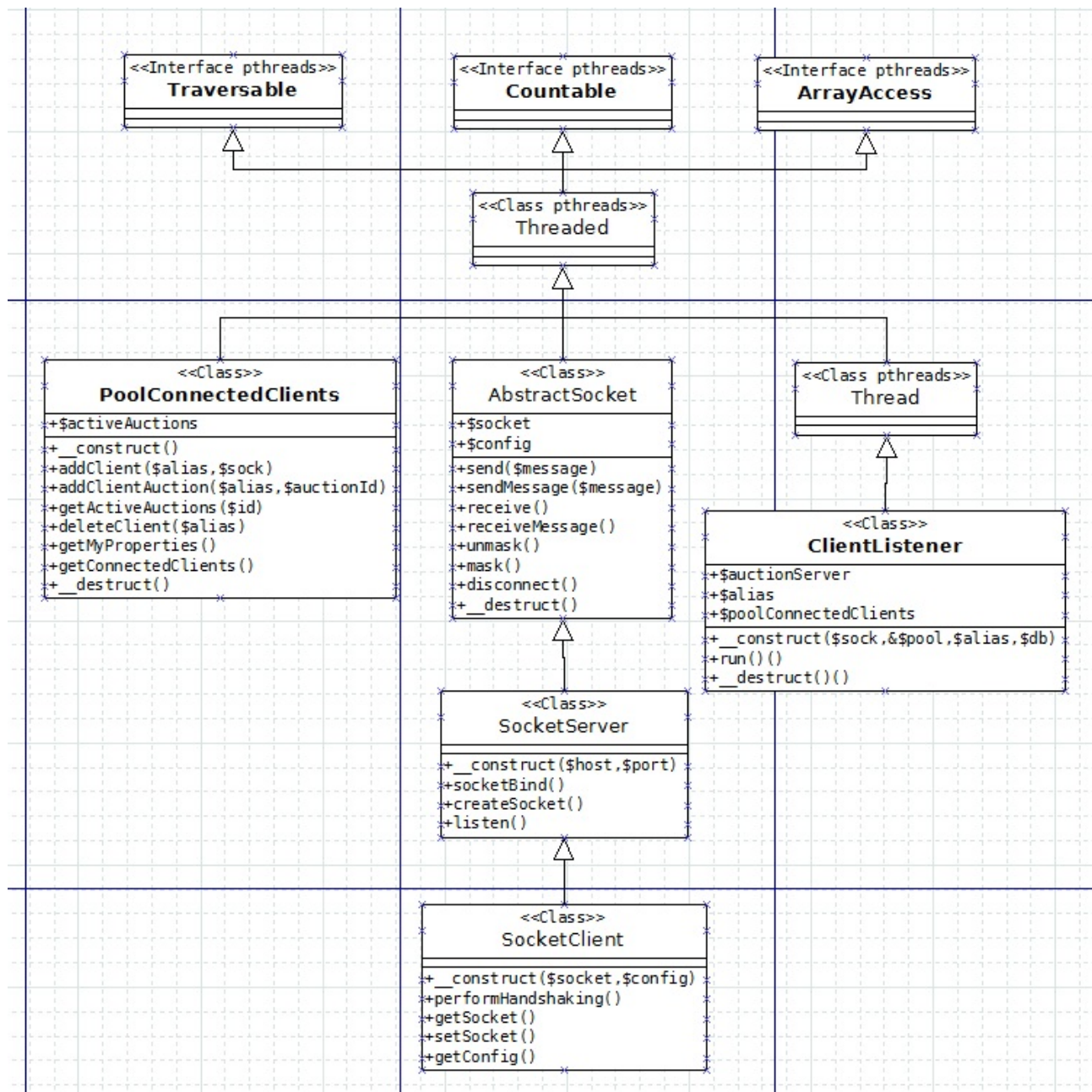


Figura 16: Diagrama de classes a nivell de connexions

A.2.2 Objectes servidors

El diagrama de la figura 17 s'orienta més als objectes que realitzen les funcions de servidors.

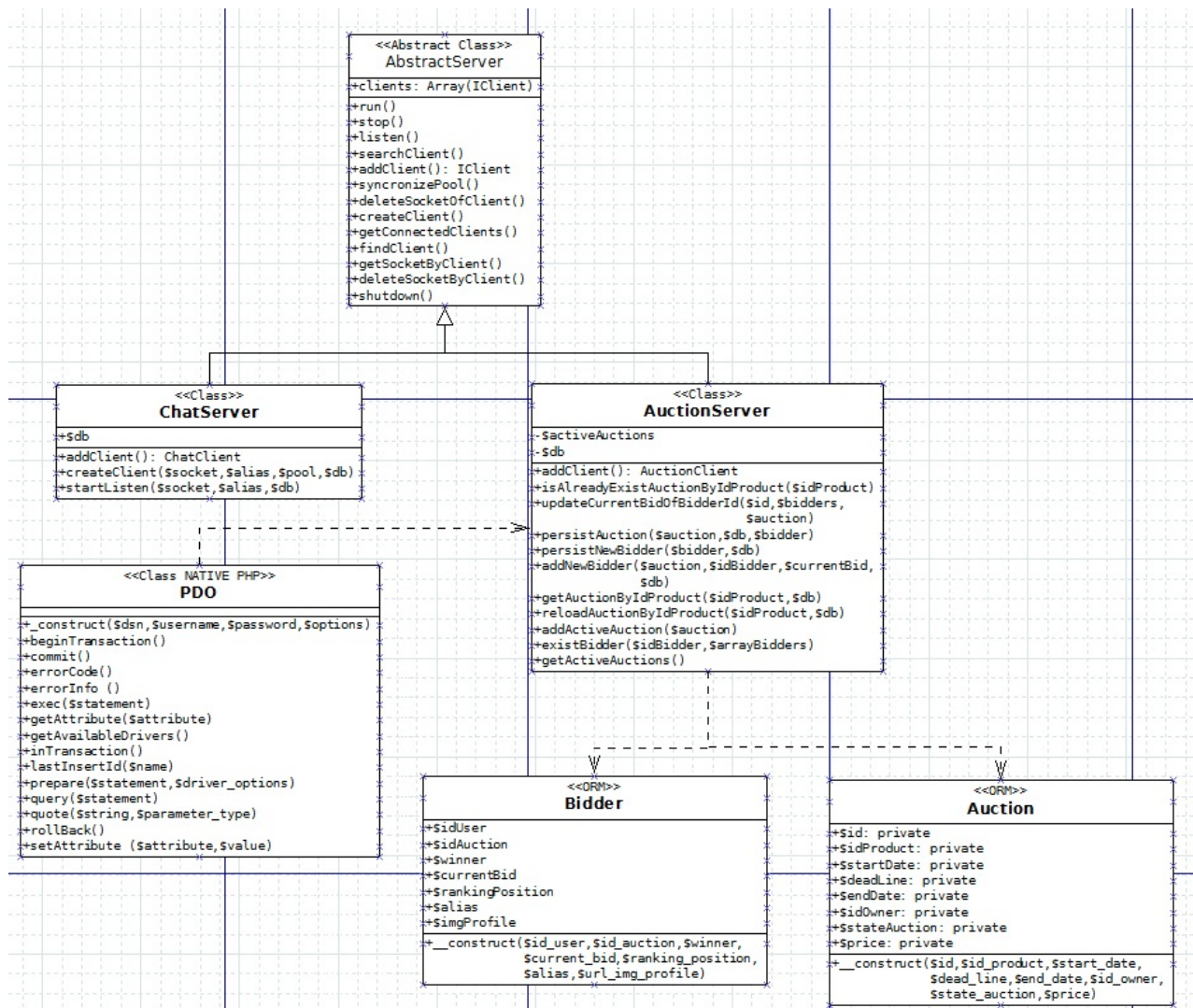


Figura 17: Diagrama de classes dels objectes servidors